# Practical Assignment - 2
# Sub: Database Technology in India

**[1]** Create Following Tables.

### Employee Table
The employee table stores information about the employees in a company.

| emp_id (Primary Key) | Unique identifier for each employee. |
|---|---|
| first_name | VARCHAR(50) |
| last_name | VARCHAR(50) |
| position | VARCHAR(50) |
| salary | DECIMAL(10,2) |
| department | VARCHAR(50) |
| hire_date | DATE |

### Product Table
The product table stores information about the products that the company offers.

| Column Name | Data Type |
|---|---|
| product_id | INT (Primary Key) |
| product_name | VARCHAR(100) |
| category | VARCHAR(50) |
| price | DECIMAL(10,2) |
| stock_quantity | INT |
| emp_id | INT (Foreign Key) |
| created_date | DATE |

**Perform following.**
1. Insert 10 Records in each table.
2. Increase the salary by 5% for employees in the 'Purchase department'.
3. Select all products whose prices are less than Rs.100.
4. List all employees in the 'Sales' department, sorted by hire_date.
5. List all employees who sell 'TV' products.
6. Delete employee data whose stock of any item is less than 50.
7. List employees whose salary is higher than the average salary of their department.

**[2]** **Create the following table with necessary constraints.**

**Table Name: Tbl_Book_Issue**

PriamryKey: B_id

| Fields | Data types | Size |
|---|---|---|
| B_id | Number | 5 |
| B_title | Varchar2 | 25 |
| Author_name | Varchar2 | 25 |
| ISBN | Number | 5 |
| Category | Varchar2 | 25 |
| Issue_Date | Date | 15 |
| Return_Date | Date | 15 |

1. Enter 5 Records.
2. Display all books whose author name is "Swami Vivekanand".
3. Display the books list which issue between 02/05/2022 to 05/10/2022.
4. Display author name who publishes more than 5 books.
5. Add new column book_price after authour _name.
6. Update column name book_category instead of category.

**[3]** **Create Following Tables.**

**Students Table**

The students table stores information about students.

| Column Name | Data Type | Description |
|---|---|---|
| student_id | INT (Primary Key) | Unique identifier for each student. |
| first_name | VARCHAR(50) | First name of the student. |
| last_name | VARCHAR(50) | Last name of the student. |
| date_of_birth | DATE | Date of birth of the student. |
| gender | CHAR(1) | Gender of the student (M for male, F for female). |
| class | VARCHAR(10) | Class or grade of the student (e.g., "BCA"). |
| admission_date | DATE | Date of admission into the school. |

## Marks Table
The marks table stores information about marks scored by students in various subjects and exams.

| Column Name | Data Type | Description |
|---|---|---|
| mark_id | INT (Primary Key) | Unique identifier for each mark entry. |
| student_id | INT (Foreign Key) | Reference to the student in the students table. |
| subject | VARCHAR(50) | Subject name (e.g., "Oracle"). |
| exam_type | VARCHAR(50) | Type of exam (e.g., "Midterm", "Final"). |
| marks_obtained | DECIMAL(5, 2) | Marks obtained by the student in the exam. |
| total_marks | DECIMAL(5, 2) | Total possible marks in the exam. |
| exam_date | DATE | Date of the exam. |

1. Insert 10 Records in each table.
2. List all students in class "BCA", ordered by their last name.
3. Calculate the average marks obtained by each student.
4. Find the total marks obtained by each student in the "Final" exam.
5. List each student's full name along with their marks in "Oracle".

**[4]** **Create following tables and perform following queries.**

**Account_Holder**

| Cust_id | Numeric(5) | Primary Key |
|---|---|---|
| Cust_name | Varchar2(25) | Not Null |
| Address | Varchar2(30) | |
| City | Varchar2(15) | |
| Account_no | Numeric(10) | Foreign Key |
| Ac_open_date | Date | |
| Mobile | Varchar2(10) | |

**Account_Master**

| Account_no | Numeric(10) | Primary Key |
|---|---|---|
| Ac_type | Char(1) | Set 'S' or 'C' |
| Tran_no | Numeric(20) | |
| Balance | Numeric(20) | |

**Perform following queries....**

1. Insert 5 records in both tables.
2. Display the Records whose account type is saving.
3. Insert new fields DOB in Account Holder table.
4. Display the Customer Name whose name start with 'J'.
5. Delete the Records whose city name is Mumbai.
6. Display all records in descending order on cust_id field form account holder table.
7. Display cust_name, account_no and balance.

## [5] Create Following Tables.

### Customer Table

The Customer table stores information about the bank's customers.

| Column Name | Data Type | Description |
| --- | --- | --- |
| customer_id | INT (Primary Key) | Unique identifier for each customer. |
| first_name | VARCHAR(50) | First name of the customer. |
| last_name | VARCHAR(50) | Last name of the customer. |
| date_of_birth | DATE | Date of birth of the customer. |
| gender | CHAR(1) | Gender of the customer (M for male, F for female). |
| address | VARCHAR(100) | Address of the customer. |
| phone_number | VARCHAR(15) | Phone number of the customer. |
| email | VARCHAR(50) | Email address of the customer. |

### Bank Account Table

The BankAccount table stores information about customer bank accounts. Each account is linked to a customer.

| Column Name | Data Type | Description |
| --- | --- | --- |
| account_id | INT (Primary Key) | Unique identifier for each bank account. |
| customer_id | INT (Foreign Key) | Reference to the customer in the Customer table. |
| account_type | VARCHAR(20) | Type of the account (e.g., "Current", "Savings"). |
| balance | DECIMAL(12, 2) | Current balance of the account. |
| opened_date | DATE | Date when the account was opened. |
| status | VARCHAR(20) | Status of the account (e.g., "Active", "Closed"). |

1. Insert 10 Records in each table.
2. List all customers whose last name is "Patel", sorted by first name in descending order.
3. Calculate the total balance for each account type.
4. Find the number of accounts for each status (e.g., Active, Closed).
5. Update the status of all bank accounts with a balance below Rs.1000 and status is "Inactive".
6. Find the maximum balance for each type of account.

**[6]** Create the following table with proper format and perform the following queries.

**Table Name : Account_master**

| Field Name | Datatype | Size | Description |
|---|---|---|---|
| Acc_no | Number | 3 | Account Number |
| Acc_name | Varchar2 | 30 | Account Holder Name |
| DOB | Date | | Date of Birth |
| Opening Date | Date | | Account opening date |
| Balance | Number | 11,2 | Current balance |
| Acc_type | Char | 1 | S For Saving, C for current |
| Gender | Char | 1 | M for male, F for female |
| Acc_status | Char | 1 | O for open , C for Close |

**Perform the following queries.**
1) Insert 10 records as needed.
2) Display all information of all account holders.
3) Display all closed account holder in ascending order.
4) Display information of all female account holders who born before 2000.
5) Count total number of Male account holder.
6) Delete all close account.
7) Add Rs. 500 in balance that have below 2000 balance.

**[7]** Create Following Tables.

**Facebook Table**
The Facebook table stores information about each user on the social media platform.

| Column Name | Data Type | Description |
|---|---|---|
| user_id | NUMBER (Primary Key) | Unique identifier for each user. |
| username | VARCHAR2(50) | Username of the Facebook user. |
| email | VARCHAR2(100) | Email address of the user. |
| date_of_birth | DATE | Date of birth of the user. |
| join_date | DATE | Date when the user joined Facebook. |
| location | VARCHAR2(100) | Location or city where the user resides. |

**Post Table**

The Post table stores information about each post made by users on Facebook.

| Column Name | Data Type | Description |
|---|---|---|
| post_id | NUMBER (Primary Key) | Unique identifier for each post. |
| user_id | NUMBER (Foreign Key) | Reference to the Facebook table for user details. |
| content | TEXT | Text content of the post. |
| post_date | DATE | Date when the post was created. |
| likes | NUMBER | Number of likes the post received. |
| shares | NUMBER | Number of times the post was shared. |

1. Enter 10 Records.
2. List all users located in "Bhavnagar", sorted by their join date.
3. Count the total number of posts made by each user.
4. Find the average number of likes per post for each user.
5. Find the most-liked post for each user.
6. Update the location for a specific user (user_id = 102) to "Bhavnagar".
7. Remove a column shares from post table.

**[8]** **Create following tables in oracle with given constraints.**

**Table name: STUDENT**

| Column Name | Data Type And Size | Constraints |
|---|---|---|
| Sid | Number(3) | Primary Key |
| Name | Varchar2(20) | Not Null |
| City | Varchar2(20) | |
| Pincode | Number(6) | Unique |

**Table name: EXAM**

| Column Name | Data Type And Size | Constraints |
|---|---|---|
| Eid | Number(3) | |
| Subname | Varchar2(20) | |
| Marks | Number(2) | |
| Sid | Number(3) | Foreign Key |

**Insert following data**

| Sid | Name | City | Pincode |
|---|---|---|---|
| 1 | Vijay | Bhavnagar | 364002 |
| 2 | Ashok | Bhavnagar | 364002 |
| 3 | Ujaas | Rajkot | 360055 |
| 4 | Ajay | Bhavnagar | 364002 |
| 5 | Pinakin | Jamnagar | 361003 |

| Eid | Subname | Marks | Sid |
|---|---|---|---|
| 101 | ORACLE | 40 | 1 |
| 101 | JAVA | 23 | 2 |
| 101 | PHP | 24 | 3 |
| 101 | ASP | 45 | 1 |
| 101 | SE | 56 | 1 |

| 6 | Bhumi | Bhavnagar | 364002 |
|---|-------|-----------|--------|
| 7 | Rajul | Rajkot | 360055 |

| 102 | JAVA | 40 | 2 |
|-----|------|----|----|
| 102 | PHP | 34 | 3 |

## Perform the Following Queries

1) Display all information of the entire STUDENT table.
2) Display all details of EXAM table.
3) List Student Name, Subject and Marks of all the students whose EID is 101.
4) List the students who never appeared any exam.
5) Count total number of students appeared in the exam 101 and 102 subjects wise.

## [9] Create Following Tables.

**Account_table:**

| Account_no | Name | DOB | Salary | Gender |
|------------|------|-----|--------|--------|
| 0011 | Rohan | 10-Mar-2001 | 10000 | Male |
| 0022 | Utsav | 12-Aprl-2001 | 15000 | Male |
| 0033 | Greeva | 12-Aug-2002 | 20000 | Female |
| 0044 | Aarvi | 12-May-2022 | 25000 | Female |
| 0055 | Misri | 05-June-2001 | 30000 | Female |
| 0066 | Shreya | 18-June-2002 | 15000 | Female |
| 0077 | Raj | 09-Sept-2003 | 45000 | Male |

## Perform following.

1) Display only Female account holder records in descending order by their name.
2) Display account holder detail that has maximum salary (using function).
3) Display the name of account holder who born between 2001-2003.
4) Display all account holder name in uppercase only (using function).
5) Delete the records whose salary is 15000 only.
6) Update only male account holder data, increase their salary by 5%
7) Display all details of account holder whose name starts with letter 'R' or U' .
8) Change field name from 'Salary' to 'Gross_salary'

**[10]** Create Following tables with appropriate data type and size, define constraint as per given instruction.

### Table Name: Student

| Name | Type(Size) | Constraint |
|---|---|---|
| Roll no. | Number(7) | Primary key |
| Name | Varchar2(30) | Not null |
| Address | Varchar2(50) | |
| City | Varchar2(20) | Only allowed 'Bhavnagar', 'Rajkot' and 'Surat' |
| Pincode | Number(6) | |
| Mobileno | Number (10) | |
| Gender | Char(1) | Must be 'M' or 'F' |
| DOB | Date | Not Null |

1) Insert minimum 10 records.
2) Display Rollno, name and city of all students.
3) Display name of students in uppercase who are living in Rajkot.
4) Display the Students whose birthday is coming in current month.
5) Display number of Male and Female students living in each city.
6) Display Student data whose name start with 'd'.
7) Display rollno between 3 to 5.